VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

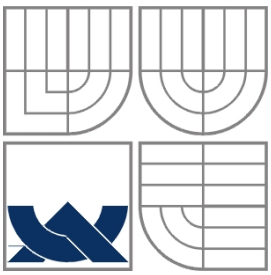
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZÍSKÁVÁNÍ ČASOVÝCH A DATOVÝCH CHARAKTERISTIK SÍŤOVÝCH SPOJENÍ

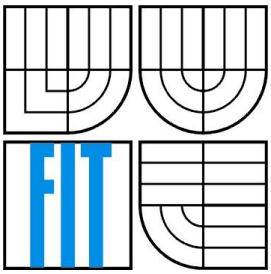
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE PETR KRAMOLIŠ
AUTHOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZÍSKÁVÁNÍ ČASOVÝCH A DATOVÝCH CHARAKTERISTIK SÍŤOVÝCH SPOJENÍ

TEMPORAL AND DATA CHARACTERISTICS ACQUISITION OF NETWORK CONNECTIONS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR KRAMOLIŠ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. ŽÁDNÍK MARTIN

BRNO 2010

Abstrakt

Tato práce se zabývá návrhem a implementací programového vybavení pro získávání časových a datových charakteristik síťových spojení. V této práci se využívá platformy flexibilního FlowMona na akceleračních kartách COMBOv2.

TEMPORAL AND DATA CHARACTERISTICS ACQUISITION OF NETWORK CONNECTIONS

Abstract

This thesis deals with design and implementation of software tools for a acquisition of temporal and data characteristics of network connections. This thesis use platform of flexible FlowMon and acceleration cards COMBOv2.

Klíčová slova

Časová značka, Netflow, Ipfix, paketový analyzátor, čas, data.

Keywords

Timestamp, Netflow, Ipfix, packet analyzer, time, data.

Citace

Petr Kramoliš: Získávání časových a datových charakteristik síťových spojení, bakalářská práce, Brno, FIT VUT v Brně, 2009

Získávání časových a datových charakteristik síťových spojení

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Martina Žádníka.
Další informace mi poskytli členové projektu Liberouter.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Petr Kramoliš
16. května 2010

Poděkování

Chtěl bych poděkovat panu Ing. Martinovi Žádníkovi za vedení mé bakalářské práce. Dále bych chtěl poděkovat všem členům Liberouteru za poskytnutí potřebných informací a spolupráci.

Obsah

1 Úvod

Rozvoj v oblasti počítačových sítí neustále sílí, ať už se jedná o navyšování přenosových rychlostí nebo o vývoj nejrůznějších zařízení, které dokáží využívat počítačových sítí. Roste počet mobilních počítačů a mobilních telefonů s možností připojení do bezdrátové sítě. Připojení k Internetu lze dneska poměrně často najít i v kavárnách nebo turistických oblastech a mnohdy zdarma. Dokonce se lze připojit k Internetu i z některých autobusů, například na trase z Brna do Rožnova pod Radhoštěm jeden takový autobus jezdí. Pro mnoho lidí je přístup k Internetu nedílnou součástí života ať už kvůli práci, komunikaci nebo zábavě. Z výzkumů vyplývá, že až čtyři z pěti lidí považují přístup k Internetu jako základní lidské právo [9].

Přes Internet a jiné počítačové sítě se v dnešní době přenáší ohromné množství dat. S rostoucím počtem přenášených dat se stává stále obtížnější monitorování síťového provozu a zajišťování bezpečnosti v síti. Může se jednat o sledování vytížení určitých síťových zařízení nebo identifikace útoků přes počítačovou síť. S vývojem a rozšiřováním Internetu se vyvíjí i počítačová kriminalita. Neustále přibývá způsobů jak zneužít počítačovou síť pro získání citlivých údajů uživatelů, likvidaci konkurence nebo prolomení bezpečnostních systémů bank. Jedná se jak o kriminální aktivitu jednotlivců, tak celých organizovaných skupin. V neposlední řadě o sobě dají někdy vědět i specializované kybernetické armády. Ať už se jedná o civilní nebo armádní sektor, je analyzování síťového provozu stále důležitější.

Analýza síťového provozu slouží také pro měření a udržování kvality služeb (Quality of Service) v počítačových a telekomunikačních sítích. Například při zahlcení sítě se vyhradí přenosové kapacity prioritním síťovým službám. Detekce aplikací pomocí sledování síťového provozu je také velmi důležitá. Jedná se o vyhledávání konkrétních znaků v síťovém provozu, pomocí kterých lze identifikovat určité aplikace. Jedná se o poměrně složitý proces, kterému musí předcházet důkladná analýza síťového provozu aplikací, které má proces detekovat. Aktivity na síti mají často více než jednu charakteristickou událost. K detekci takovýchto aktivit je třeba prohledat události síťového provozu a pokusit se rozeznat události, které spolu souvisí. Tento proces se nazývá korelace a je často požadováno, aby korelace probíhala v reálném čase.

Existuje celá řada nástrojů umožňující měření a analyzování síťového provozu. Nejdostupnější pro běžného uživatele je softwarové vybavení umožňující měření a analýzu síťového provozu na jednotlivých rozhraních počítače. Jedná se o paketové analyzátory, které zobrazují podrobné informace o každém zachyceném paketu, používané často i při vývoji síťových aplikací. Dále několik protokolů určených pro správu nejrůznějších zařízení může přenášet informace o síťovém provozu daného zařízení. Pro spravování velkých sítí existují speciální směrovače, které dokáží zasílat podrobné informace o datech procházejících skrz ně. Podobné měření mohou obstarávat i

specializované sondy. Sondy se dají umístit na jakékoli místo v síti a pro uživatele sítě jsou nedetekovatelné. Sondy jsou určené na měření rychlých sítí, jsou tedy vhodné například pro poskytovatele připojení. Je možné zapojit do sítě několik sond a shromažďovat naměřené informace na jednom stroji, který tak podá uživateli podrobnější a rozsáhlejší přehled o pohybu dat v síti. Nejedná se ovšem o levnou záležitost.

Pro měření síťového provozu existuje několik standardů a protokolů. Jedná se například o protokoly pro posílání naměřených dat ze sond a směrovačů na zařízení zpracovávající data. Velmi rozšířený protokol Netflow verze pět, který umožňuje zasílání pevně určených informací o síťovém provozu, je pomalu nahrazován novějšími protokoly, které umožňují zasílaní uživatelem specifikovaných dat. Z důvodů rychlého rozvoje a pro rozšíření možností měření se při vývoji nových zařízení a protokolů klade důraz na flexibilitu.

Dokument je rozdělen na několik částí. Po úvodu následuje kapitola věnována teoretickému rozboru. V této je čtenář seznámen s měřením časových a datových charakteristik síťových spojení. Jsou zmíněny zařízení a aplikace umožnující tato měření. Dále se popisuje rozpoznávání datových toků a jejich indexace. Poté je čtenář seznámen se získáváním přesných časových značek a datových charakteristik síťových spojení. Kapitola je uzavřena exportováním informací na kolektor a knihovnou Pcap. Následuje kapitola zabývající se implementací nástroje pro měření časových a datových charakteristik síťových spojení. Je zde popsán projekt Liberouter, ve kterém se komponenta vyvíjela, a návrh samotné aplikace. Dále je postupně popsána implementace jednotlivých částí a testování celého nástroje. V samotném závěru je zhodnocení dosažených výsledků.

2 Teoretický rozbor

2.1 Měření síťových charakteristik

Existuje mnoho způsobů jak měřit síťové charakteristiky. Liší se především podle potřeb měření síťových charakteristik jednoho konkrétního zařízení nebo veškerého provozu procházejícího například směrovačem v dané síti. Nástroje umožňující měření na konkrétním zařízení vyžadují, aby byl na zařízení spuštěn daný nástroj nebo kompatibilní agent. Uživatel je tedy limitován zařízeními, na které má přístup, nebo na zařízení, kde patřičné softwarové vybavení obstará někdo jiný. Příklad několika nástrojů umožňujících zmíněný typ měření [25].

2.1.1 Paketové analyzátory

Paketové analyzátory zachytávají síťový provoz na daném zařízení a v případě potřeby dekódují pakety. Paketové analyzátory nejčastěji pracují s jednotlivými pakety zvlášť a na rozdíl od sond nezařazují pakety do datových toků. Jejich účel spočívá v poskytnutí uživateli co nejlepšího nástroje pro podrobnou analýzu paketů. Paketové analyzátory se často používají při vývoji síťových aplikací, přičemž si uživatel může zkontrolovat, jestli jsou pakety odesílány a zda jsou ve správném formátu. Jeden z nejznámějších analyzátorů je Wireshark, který se běžně používá při analyzování a ladění síťových aplikací. Často se taky používá ve školách v počítačových laboratořích, protože má velmi jednoduché uživatelské rozhraní a uživatel se s ním naučí velice rychle pracovat [11].

2.1.2 Simple Network Management Protocol

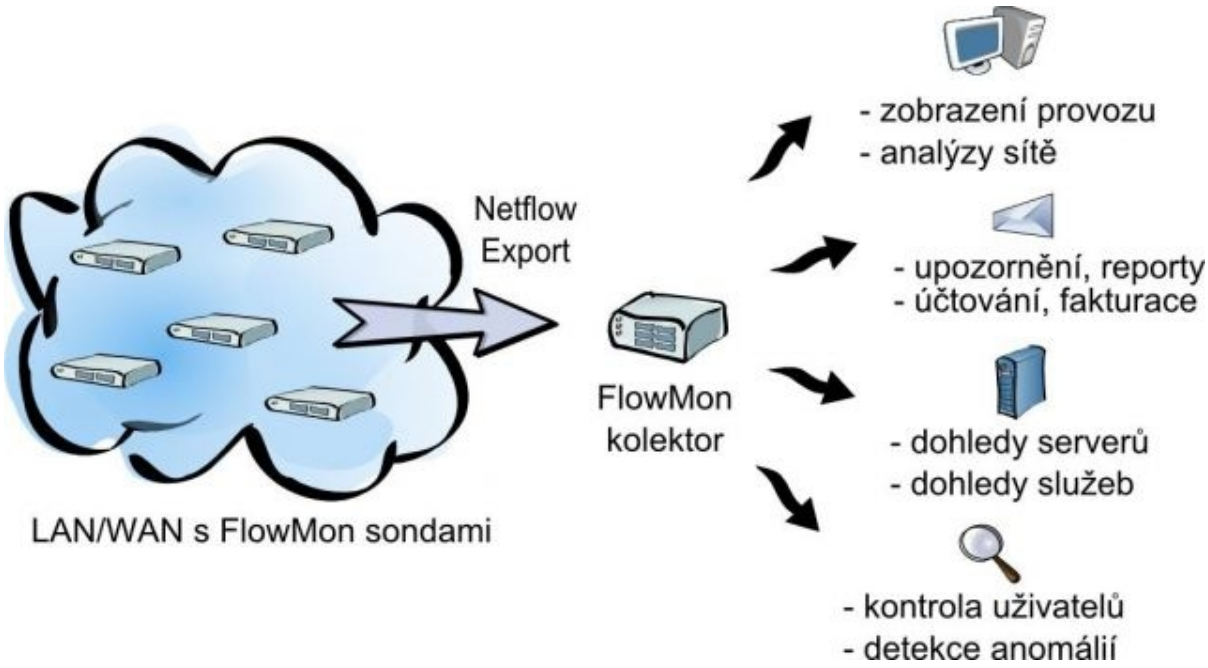
Simple Network Management Protocol (SNMP) je široce rozšířený protokol pro spravování nebo sledování nejrůznějších zařízení. SNMP pracuje na komunikaci mezi klientem a serverem. Protokol byl původně určen pro správu sítě, ale jeho využití se rozšiřuje i do ostatních odvětví informatiky a průmyslu. SNMP umožňuje zasílaní informací mezi monitorovanými zařízeními a kontrolním zařízením, které klade monitorovaným zařízením nejrůznější požadavky. Zasílané zprávy můžou přenášet například informace o překročení mezních údajů při měření nebo výpadku určitého zařízení [12 , 18].

2.1.3 Windows Management Instrumentation

Windows Management Instrumentation (WMI) je systém vyvinutý firmou Microsoft a slouží obdobně jako SNMP. Můžeme jej najít na operačních systémech Windows. Od operačního systému Windows 2000 a novějších systémů je software předinstalován. WMI je kompatibilní s SNMP a s podobnými řídícími systémy [16].

2.1.4 Sondy a směrovače

Měření veškerého síťového provozu se provádí nejčastěji pomocí zařízení umístěných na výstupu z lokální sítě. Tento úkol mohou obstarávat některé směrovače nebo speciální sondy. Měření se provádí zpravidla tak, aby samotné měření a sondy nemohl detekovat uživatel sítě. Na sondách a směrovačích se zachytává každý paket a nejčastěji se pakety přiřazují do datových toků. Následně se vypočítají potřebné statistiky a data se exportují na zařízení zvané kolektor. Kolektor sbírá data od jednoho nebo několika exportujících zařízení a zobrazuje důležité informace a statistiky uživateli. Rozmístění sond v síti tak, aby podávaly co nejlepší informace o síťovém provozu, může být komplikované. Problematikou rozmístění sond v síti se zabývá například dokument Optimal positioning of active and passive monitroring devices [5]. Získané údaje podávají poměrně detailní přehled o síťovém provozu. Při tomto způsobu měření se posílají jen statistiky o daných datových tocích a několik doplňujících informací. Neposílají se samostatné pakety a jejich obsah, uživatel tedy nemá možnost podrobně analyzovat jednotlivé pakety. Zná pouze odkud kam se poslalo určité množství dat pomocí určitého síťového protokolu.



Obrázek 2.1 Netflow sondy (převzato z [27])

2.2 Datové toky

Síťové statistiky a charakteristiky se měří nejčastěji pro jednotlivé datové toky zvlášť. Jako datový tok se označuje seskupení několika paketů, které mají shodné určité údaje. Existuje několik způsobů, jak odlišovat jednotlivé datové toky. Často se jedná o tuto pětici údajů:

- zdrojová IP adresa,
- cílová IP adresa,
- zdrojový port,
- cílový port,
- protokol.

Další vhodné údaje pro odlišování toku:

- IP protokol,
- ToS (Type of Service).

V informacích o datových tocích najdeme pouze informaci, kolik bylo přeneseno paketů, ale nemáme přístup k daným paketům a datům, která přenášely. Ukládaní a exportování takovýchto dat by bylo na rychlejších sítích nemožné.

2.3 Indexace

Při pořizování charakteristik na rychlých sítích je velice důležitá rychlost zpracování jednotlivých paketů a způsob přiřazení paketů k daným tokům. Využívá se hašovacích funkcí, které převedou údaje pro odlišování jednotlivých datových toků do jediného čísla, označovaného jako haš nebo otisk. Tato haš se dále využívá při přiřazování paketu k toku a identifikaci jednotlivých toků. Může zde nastat problém, že je vygenerována stejná haš pro různé datové toky. Zmíněný případ je nazýván jako kolize haše.

2.3.1 Hašovací funkce

Hašovací funkce je algoritmus, který převádí vstupní data na jedno číslo. Nejčastěji se jedná o libovolně dlouhý textový řetězec převáděný na číslo, které má přesně definovanou maximální hodnotu, a ta se nesmí přesáhnout. Výsledné číslo se nazývá haš nebo otisk a je používáno k rychlejšímu vyhledávání dat než by bylo možné s například padesáti znakovým řetězcem.

Matematicky je vyloučeno, abychom z libovolně dlouhého řetězce dostali pokaždé jiné číslo s omezenou maximální velikostí. Pokud se stane, že hašovací funkce vygeneruje dvě shodné haše pro různé vstupní hodnoty, nastala takzvaná kolize haší. Jakákoli kolize haše je nepříjemná a dokáže buďto zpomalit program nebo dokonce způsobit zásadní chybu. Například došlo-li by ke kolizi haše na proxy serveru mohlo by se stát, že se data pošlou špatnému uživateli a původní uživatel by se ke svým datům již nikdy nedostal. Hašovací funkce jsou asi nejvíce využívané v databázových systémech. Hašovací funkce by měla splňovat následující vlastnosti:

- převedení libovolně dlouhého vstupu na stejně dlouhý otisk,
- z výsledné haše je nemožné rekonstruovat původní data,
- hašovací funkce by měla dosahovat co nejmenšího počtu kolizí,
- drobná změna vstupu by se měla výrazně projevit na otisku [24].

2.3.2 Cuckoo haš

Cuckoo haš neboli kukačková haš využívá dvou hašovacích funkcí. Pro každý prvek jsou vygenerovány dvě haše. Prvek je vložen na pozici jedné z haší a případný prvek, který byl dříve na dané pozici je přesunut na pozici své druhé haše. Tento princip se opakuje tak dlouho, dokud není nalezena pozice pro všechny prvky nebo se vyhledávání dostane do nekonečné smyčky. Výhoda cuckoo haše spočívá v době, za jakou se vyhledá daný prvek. Maximálně je nutné zkontrolovat dvě položky a to se provede v nejhorším případě za konstantní čas [22].

2.4 Časové značky

Jedním z nejdůležitějších informací při měření síťového provozu jsou časové značky. Udávají co nepřesnější časové informace o příchodu paketu, který se po přidělení časové značky zpracovává a případně přiděluje k datovému toku. Pro každý datový tok se přiřazují alespoň dvě časové značky a to pro začátek a konec datového toku (příchod prvního a posledního paketu). Nástroje pracující s jednotlivými pakety přiřazují časovou značku každému paketu. Při rychlostech okolo deseti gigabitů za sekundu se může vyskytnout i dvacet milionů paketů za sekundu a pro zachycení časových značek, které by nám spolehlivě odlišovaly příchody paketu, potřebujeme přesnost minimálně nad padesát nanosekund. Pro přidělení přesné datové značky musí mít dané zařízení přístup k co nejpřesnějšímu času, zde lze využít například PPS, GPS a NTP.

2.4.1 Pulse per Second

Pulse per Second (PPS) je elektrický signál, který velice přesně značí začátek každé sekundy. Přesnost se liší podle použitého zařízení ale pohybuje se v rozmezí od milisekund až po několik nanosekund. Důležité je, že PPS nenese informace o aktuálním čase, ale pouze indikuje začátek sekundy. To znamená, že pomocí PPS můžeme udržovat přesný čas, ale nemůžeme čas nastavit [17].

2.4.2 Global Positioning System

Global Positioning System (GPS) je systém schopný určit přesnou polohu a čas na Zemi. GPS provozuje Ministerstvo obrany Spojených států amerických a s omezenou přesností je volně přístupné všem civilním uživatelům. Jedná se o družicový systém, který dokáže využívat až třicet dva družic. Každá družice obsahuje několik atomových hodin a pro zajištění přesnosti se využívá i Einsteinova principu relativity. Každá družice vysílá pravidelně signál nesoucí informace o přesném čase, kdy byl signál vyslán, informace o orbitu družice a přibližné informace o stavu a pozici ostatních družic. Zařízení, které využívá systému GPS, potřebuje zachytit signál z minimálně čtyř satelitů, aby bylo zařízení schopné vypočítat přesný čas a pozici. Po celém světě je nespočet zařízení využívajících GPS, ať už se jedná o navigační systémy, nebo zařízení detekující epicentra zemětřesení pomocí přesného času. GPS zařízení dosahují přesnosti od jedné mikrosekundy až po 50 nanosekund [6].

2.4.3 Network Time Protocol

Network Time Protocol (NTP) je protokol pro synchronizaci času počítačových hodin po síti. Jedná se o distribuci co nejpřesnějšího času přes paketové sítě pomocí specializovaných serverů. Protokol je navržený tak, aby odolával rozdílnému zpoždění při doručování paketů v síti. Dosahovaná přesnost protokolu je od deseti milisekund až po dvě stě mikrosekund. Pro výpočet přesného času se využívá několika zpráv z různých NTP serverů, ze kterých se vypočítá co nejpřesnější čas [21].

2.5 Datové charakteristiky

Datové charakteristiky k jednotlivým paketům nebo datovým tokům jsou získávány ze samotného paketu. Nejdříve je nutné identifikovat protokoly, pomocí kterých jsou data přenášena. Standardně se identifikuji základní protokoly jako ethernet, IP, TCP a UDP. V IP protokolu nalezneme například IP adresy příjemce, odesílatele a informaci o dalším protokolu, nejčastěji UDP nebo TCP. Dále se počítá množství paketů pro jednotlivé toky a velikost toku v bajtech. Nejzákladnější datové charakteristiky:

- zdrojová IP adresa,
- cílová IP adresa,
- zdrojový port,
- cílový port,
- verze IP protokolu,
- protokol (UDP, TCP...),
- ToS (Type of Service),
- TCP příznaky.

2.6 Export

Exportováním zde rozumíme přenesení naměřených dat z měřících zařízení - sond, směrovačů a podobně (hromadně nazývaných exportéry), na zařízení, která data ukládají a umožňují uživateli přístup k naměřeným hodnotám a nejrůznějším statistikám. Exportují se informace o jednotlivých datových tocích a obecné informace o síťovém provozu jako celkový počet zachycených paketu od minulého exportu dat. Datové toky se exportují při jejích ukončení nebo při uplynutí časového intervalu u aktivních toků. Obecně se jedná o paket s několika záznamy datových toků opatřených hlavičkou s obecnými informacemi, odesílaný nejčastěji protokolem UDP, někdy se využívá i u protokolu TCP.

2.6.1 Netflow verze pět

Pro samotný formát exportovaných dat existuje několik protokolů, mezi které patří například hojně využívaný protokol Netflow verze pět [8]. Jedná se o protokol s pevně stanovenými prvky, které se exportují, to znamená, že je uživatel při exportování dat omezen na dané prvky. Hlavička Netflow paketu je popsána tabulkou 2.1.

Bajty	Obsah	Popis
0 – 1	version	Verze Netflow protokolu
2 – 3	count	Počet exportovaných toků v tomto paketu
4 – 7	sys_uptime	Čas uplynulý od nastartování exportovacího zařízení
8 – 11	unix_secs	Počet uplynulých sekund od roku 1970 UTC.
12 – 15	unix_nsecs	Zbývající čas v nanosekundách od roku 1970 UTC.
16 -19	flow_sequence	Počet všech zachycených datových toků.
20	engine_type	Typ zařízení
21	engine_id	Číslo rozhraní na zařízení
22 – 23	sampling_interval	První dva bity udávají druh vzorování, zbytek udává hodnotu vzorkování.

Tabulka 2.1 Hlavička Netflow verze pět paketu

Pro každý datový tok se exportují následující informace:

Bajty	Obsah	Popis
0 – 3	srcaddr	Zdrojová IP adresa
4 – 7	dstaddr	Cílová IP adresa
8 – 11	nexthop	IP adresa dalšího hopu
12 – 13	input	SNMP index vstupního rozhraní
14 – 15	output	SNMP index výstupního rozhraní
16 - 19	dPkts	Počet paketů v datovém toku
20 – 23	dOctets	Počet bajtů všech paketů ve třetí (síťové) vrstvě
24 – 27	first	Čas uplynulý od nastartování systému při zaznamenání startu toku.
28 – 31	last	Čas uplynulý od nastartování systému při zaznamenání konce toku.
32 - 33	srcport	Číslo zdrojového portu
34 – 35	dscport	Číslo cílového portu
36	pad1	Nevyužito (nulová hodnota)
37	tcp_flags	Sjednocení všech TCP flagů
38	prot	IP protocol type (TCP, UDP...)
39	tos	Type of Service (ToS)
40 – 41	src_as	Hodnota zdrojového autonomního systému
42 - 43	dst_as	Hodnota cílového autonomního systému
44	src_mask	Maska zdrojové IP adresy
45	dst_mask	Maska cílové IP adresy
46 - 47	pad2	Nevyužito (nulová hodnota)

Tabulka 2.2: Záznam Netflow verze 5

2.6.2 Internet Protokol Information Export

Pro získání větší flexibility byl navržen protokol Netflow verze 9, který byl následně rozšířen protokolem Internet Protokol Information eXport (IPFIX), informace o IPFIX jsou čerpány z [1,2,7, 26]. Protokoly pracují na bázi šablon, kterými si uživatel může definovat libovolné prvky, které potřebuje exportovat. Při tvorbě šablon si uživatel může vybrat z předdefinovaných datových typů nebo si může definovat své vlastní. Lze exportovat i několik datových toků s rozdílnými šablonami v jednom paketu. Každý IPFIX paket obsahuje následující hlavičku:

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	**6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1**
Version Number	Length
Export Time	
Sequence Number	
Observation Domain ID	

Tabulka 2.3: Hlavička IPFIX paketu

Version Number

Verze formátu záznamu v daném paketu. Aktuální hodnota je 0x000a.

Length

Celková velikost IPFIX zprávy, obsahuje i velikost samotné IPFIX hlavičky.

Export Time

Počet uplynulých sekund od roku 1970 po okamžik kdy je zpráva exportována.

Sequence Number

Zbytek všech odeslaných datových záznamů daným zařízením po celočíselném dělení číslem 2^32. Sequence Number se nezvětšuje v případě odesílaní šablon a podobně. Pomáhá kolektoru zjistit, jestli se neztratil nějaký datový záznam.

Observation Domain ID

Identifikátor pro dané exportovací zařízení. Hodnota by měla být alespoň lokálně unikátní.
Umožňuje kolektoru identifikovat zdroj dat.

Za hlavičkou může následovat několik druhů dat. Asi nejpodstatnější jsou definice šablon a samotný záznam s exportovanými daty. Před každým blokem dat je uvedena hlavička (Set Header) identifikující obsah dat a jejich velikost.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Set ID	Length

Tabulka 2.4: Set Header

Set ID

Set ID je číslo identifikující následující data. Například číslo dvě označuje definici šablony a čísla nad dvě stě padesát pět označují data popsané šablonou daného čísla.

Length

Celková velikost následujícího datového bloku a případného zarovnání. Velikost samotné hlavičky (Set Header) je také započítána.

To co dělá IPFIX flexibilním, je definice vlastních prvků, které chceme exportovat. To se provádí definováním šablon. Pro každou šablonu se definuje lokálně unikátní číslo, které se později používá v hlavičce (Set Header). Dále se udává počet položek, jejich typ a velikost. V případě definice vlastního typu se doplňuje ještě identifikační číslo společnosti, která daný typ využívá. Šablona má následující strukturu:

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5		6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Template ID		Field Count
E	Information Element	Field Length
...		
E	Information Element	Field Length
Enterprise Number		
...		

Tabulka 2.5: Šablona IPFIX

Template ID

Pro každou vytvořenou šablonu se přiřazuje lokálně unikátní identifikační číslo. Čísla až do dvě stě padesáti pěti jsou rezervována. Zbylá čísla jsou dostupná a slouží k přiřazení datových záznamů k patřičné šabloně.

Field Count

Počet prvků v dané šabloně.

E (Enterprise bit)

Enterprise bit určuje, zda je následující prvek standardně definovaný prvek nebo zda se jedná o prvek definovaný uživatelem (společností). V případě, že se jedná o prvek definovaný uživatelem, za definicí prvku a jeho velikosti musí následovat Enterprise Number.

Information Element

Identifikační číslo pro daný prvek. Hodnoty pro vestavěné typy lze nalézt například v RFC5102 a pro uživatelem definované si hodnotu volí sám uživatel.

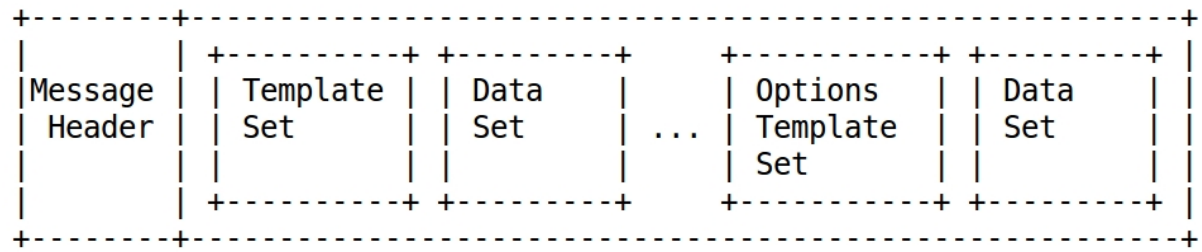
Field Length

Field Length udává velikost daného prvku, lépe řečeno patřičného datového typu.

Enterprise Number

Identifikuje společnost využívající daný prvek.

Datové záznamy se vyplňují podle patřičných šablon, musí se dodržet stejné pořadí a velikost prvku, jak je uvedené v šabloně. Před datovým záznamem musí být uvedena hlavička (Set Header) s číslem šablony a správnou délkou. Hlavičku stačí uvést před první datový záznam v případě, že následují stejné datové záznamy.



Obrázek 2.2 Příklad IPFIX paketu (převzato z [2])

2.7 Pcap

Pcap (Packet CAPture) je knihovna umožnující zachytávání paketů pro Unixové systémy a pro systémy Windows existuje portovaná verze. Při psaní aplikace využívající Pcap knihovnu se postupuje podle následujících bodů:

- výběr rozhraní,
- inicializace Pcap,
- nastavení filtrů,
- cyklus zachytávaní paketů,
- ukončení.

Nejdříve je třeba vybrat síťové rozhraní, ze kterého se budou zachytávat pakety. V případě potřeby lze využívat několik síťových rozhraní najednou. Pro linuxové systémy lze zadat například eth0, wlan0 a podobně. Následně se inicializuje Pcap knihovna a jednotlivým rozhraním se přiřazují identifikátory, podobné jako se používají při souborových aplikacích. Pokud nechceme zachytávat veškerý provoz na daných rozhraních, lze nastavit filtry, které budou propouštět jen požadovaný síťový provoz. U filtrů lze nastavit velké množství parametrů, od velikosti paketu a čísla portů až po multicastové pakety. Následuje hlavní cyklus Pcap knihovny zachycující jednotlivé pakety. Při každém zachyceném paketu, který vyhovuje nastavenému filtru, se zavolá uživatelem definovaná funkce. V dané funkci může uživatel provádět veškeré potřebné zpracování paketů, nejčastěji výpis paketů uživateli nebo zápis paketů do souboru. Po ukončení zachytávání paketů je potřeba ještě uzavřít identifikátory jednotlivých rozhraní [23].

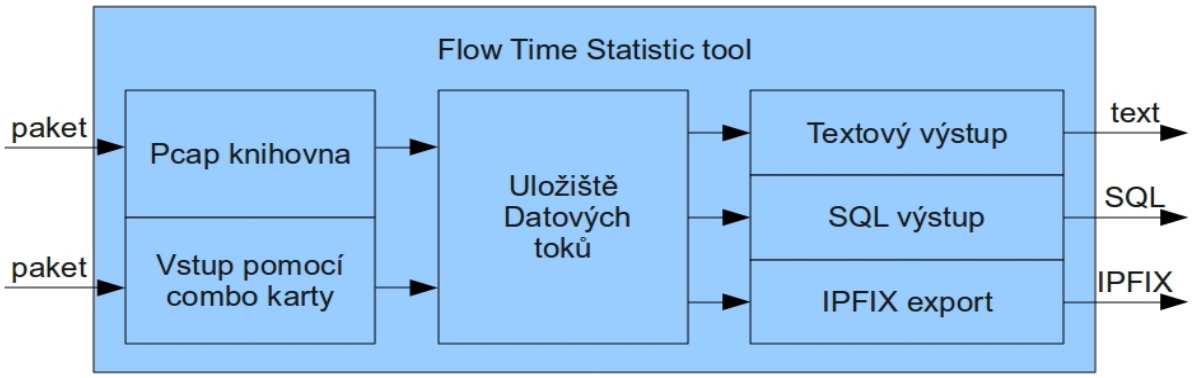
3 Implementace

3.1 Flow time statistic tool

Flow Time Statistic tool (FTS) je nástroj, který vznikl v rámci výzkumného projektu liberouter. Projekt se zabývá vývojem hardwarově akcelerovaných aplikací pro vysokorychlostní sítě. Jedná se o vývoj síťových karet s programovatelnou jednotkou FPGA, jejich firmware, ovladače a další software. Karty jsou označovány jako COMBO karty a pro jejich správnou funkci se nahrává několik odlišných firmware v projektu nazývaných jako designy. Všechen firmware byl vytvořen pomocí jazyka VHDL. Jsou zde tři vývojové skupiny NIFIC, NetCOPE a FFlowMon. NIFIC vyvíjí hardwarově akcelerované filtrovaní a směrování paketů v síti s rychlostí 10Gbps. NeCOPE vytváří platformu pro rychlé vytváření síťových aplikací pro COMBO karty. FFlowMon vyvíjí sondu pro sledování síťového provozu za pomocí COMBO karet a právě v rámci FFlowMona vznikl nástroj FTS. Na vývoji FTS udělali obrovský kus práce Tomáš Koníř a Radek Krejčí. Můj hlavní úkol je implementace IPFIX exportéru a paketových statistik. Dále úpravy programu na základě výsledků testů provedených testovací skupinou, jako například dopisování chybějících informací do nápovědy k programu nebo oprava chyb v programu. Flow time statistic tool je nástroj měřící nejrůznější časové statistiky síťového provozu. Nástroj pracuje na všech rozhraních podporovaných knihovnou Pcap, dále pracuje na COMBO kartě s FFlowMon designem timestats+L7. FTS umožňuje několik druhů výstupů a to textový výstup, zápis pomocí SQL jazyka a exportování záznamů pomocí IPFIX protokolu. Dále FTS umožňuje vypisovat čas příchodu a velikost každého paketu zvlášť v textovém režimu.

3.1.1 Návrh FTS

FTS je navržen pro zachycení paketů pomocí knihovny Pcap nebo COMBO karty. Zachycené pakety se analyzují a přiřadí se jim haš. Následně se pomocí haše dohledává datový tok, ke kterému paket patří. Pokud se nalezne, je paket k datovému toku přidán, jinak se vytvoří nový datový tok s daným paketem. Datové toky jsou ukládány do fronty, kde se jim počítá aktivní a neaktivní časový limit. Pokud datovému toku vyprší jeden z limitů, je předán textovému, SQL nebo IPFIX výstupnímu rozhraní. Při textovém nebo SQL výstupu se data vypisují na standardní výstup nebo do souboru. IPFIX výstup exportuje data na kolektor.



Obrázek 3.1 Flow Time Statistic tool

Při použití Pcap knihovny se musí počítat haš, v případě využití COMBO karty haš počítá samotná karta. Pro výpočet haše byla zvolena volně dostupná hašovací funkce od firmy PostgreSQL Database Management System. Tato haš byla v rámci projektu Liberouter použita už dříve a měla dobré výsledky, proto byla vybrána i pro FTS.

3.1.2 Záznamy datových toků

Pro uchovávání informací zachycených pomocí Pcap knihovny nebo COMBO karty se využívá struktura *flow_t* pro každý datový tok. Struktura *flow_t* obsahuje kromě informací o datovém toku i haš a pomocné proměnné umožňující práci se strukturou pomocí fronty. Struktura *flow_t* uchovává následující informace o každém toku:

- začátek datového toku v mikrosekundách,
- konec datového toku v mikrosekundách,
- počet bajtů v datovém toku,
- počet paketů v datovém toku,
- verze IP protokolu,
- protokol (UDP, TCP...),
- zdrojová IP adresa (IPv4 nebo Ipv6),
- cílová IP adresa (IPv4 nebo Ipv6),
- zdrojový port,
- cílový port,
- typ Icmp,
- maximální mezipaketová mezera,
- minimální mezipaketová mezera,
- součet všech druhých mocnin mezipaketových mezer.

3.1.3 Pcap rozhraní

Pro zařízení nedisponující COMBO kartou se využívá Pcap knihovny. Informace zachycené pomocí Pcap knihovny se převádějí do formátů obdobných ve struktuře *flow_t*. Je nutné převést časovou značku, která se v Pcap knihovně ukládá v sekundách a mikrosekundách na celkový počet mikrosekund, s nimiž pracuje FTS. Dále je potřeba rozeznat protokoly, pomocí kterých se data přenáší, zejména protokoly IP verze 4, IP verze 6, TCP , UDP, a zjistit zda se nejedná o paket z virtuální sítě. Ve chvíli, kdy jsou známé protokoly, stačí překopírovat potřebná data do struktury *flow_t*. Na rozdíl od využití COMBO karty, která obstarává výpočet haše pro každý paket, se při využívaní Pcap musí haš počítat dodatečně. Haš se počítá z těchto údajů:

- zdrojový port,
- cílový port,
- zdrojová IP adresa,
- cílová IP adresa,
- verze IP protokolu,
- protokolu (UDP, TCP …).

V případě ICMP paketu se používá pro výpočet haše typ icmp paketu.

3.1.4 Flowmon rozhraní

V případě využití COMBO karty odvede většinu práce samotná karta. Stačí rozeznat verzi IP protokolu, protože je potřeba vědět, kolik dat je třeba kopírovat pro získání IP adresy. Haš nemusíme počítat, protože karta počítá haš sama. Jedinou komplikací je výpočet časové značky, která udává čas v počtu sekund a ve zlomcích sekund. Zde se musí obě hodnoty převést na mikrosekundy. Sekundy se převedou jednoduše. Pro převod zlomků sekund se postupuje následovně: vypočítá se kolik zlomků sekund se rovná jedné mikrosekundě a poté stačí vydělit všechny zlomky sekundy počtem zlomků sekund v mikrosekundě, čímž dostaneme počet mikrosekund.

3.1.5 Časové statistiky

Pro každý datový tok se počítají časové statistiky o mezipaketových mezerách:

- průměrná mezipaketová mezera,
- maximální mezipaketová mezera,
- minimální mezipaketová mezera,
- rozptyl mezipaketových mezer.

Průměrná mezipaketová mezera se počítá na základě celkové doby datového toku a počtu mezipaketových mezer v toků:

$$P = \frac{k - z}{p - 1} \quad \text{Vzorec 2.1}$$

P – průměrná mezipaketová mezera

k – konec datového toku

z – začátek datového toku

p – počet paketů v datovém toku

Minimální a maximální mezipaketová mezera se určují porovnáváním každé mezipaketové mezery a ukládají se extrémy. Pro výpočet rozptylu se využívá součtu všech druhých mocnin mezipaketových mezer ukládaných v struktuře *flow_t*.

$$R = \sqrt{\frac{1}{p-1} * (S - \frac{(k-z)^2}{p-1})} \quad \text{Vzorec 2.2}$$

R - Rozptyl mezipaketových mezer

k – konec datového toku

z – začátek datového toku

p – počet paketů v datovém toku

S - součet všech druhých mocnin mezipaketových mezer

$$S = \sum d_i^2 \quad \text{Vzorec 2.3}$$

S - součet všech druhých mocnin mezipaketových mezer

d_i – mezipaketová mezera daného datového toku

3.1.6 Textový výstup

Jedná se o základní výstup FTS. Naměřené hodnoty se vypisují přímo na standardní výstup. Každý datový tok se vypisuje při ukončení datového toku nebo v pravidelných intervalech u aktivních toků. Konec datového toku lze v některých případech identifikovat v paketech, například FIN segment v TCP komunikaci. Pro datové toky, u kterých nejsme schopni identifikovat konec v samotné komunikaci, je nastaven časový limit pro neaktivní toky (inactivity timeout). Pokud není obdržen nový paket patřící k danému datovému toku do vypršení tohoto časového limitu, tak je datový tok označen za ukončený a vypsán na výstup. V případě dlouho trvajících datových toků je nastaven časový limit pro aktivní toky (active timeout). Po uplynutí tohoto limitu u stále neukončeného datového toku se daný tok vypíše na výstup a začne se od začátku počítat aktivní časový limit. Každý datový tok je vypsán na jeden řádek, přičemž se vypisují pouze naměřené hodnoty bez popisu, o jaká data se jedná. U FTS existuje přepínač *-H* pro vypsaní identifikátorů jednotlivých hodnot na první řádek před záznamy. Dále je zde přepínač *-v file* pro zápis výstupů do zadaného souboru místo na standartní výstup.

3.1.7 Výstup SQL

Při volbě SQL výstupu, tedy pomocí standardního dotazovacího jazyka používaného pro práci s daty v relačních databázích, se nevypisují samostatná data, ale celé příkazy jazyka SQL pro ukládání dat do databáze. Výpis jednotlivých datových toků je řešen pomocí rozeznání konce toku a časových limitů, stejně jako textový výstup. Pro každý datový tok je při výpisu vygenerován samostatný příkaz pro uložení veškerých jeho dat do databáze. Pro tvorbu samotných tabulek se zde využívá přepínač *-H,* který v SQL módu vypíše příkazy na vytvoření tabulky pro ukládaní dat datových toků. I SQL mód lze přesměrovat z výpisu na standardní výstup na zápis do souboru pomocí přepínače *-v file*.

3.1.8 IPFIX export

Místo výstupu na lokálním stoji pomocí textového nebo SQL režimu lze zvolit režim pro exportování záznamů na vzdálený počítač pomocí protokolu IPFIX. Datové toky, které by se pomocí textového nebo SQL režimu vypsaly, jsou ukládány do struktury *ipfix_pkt_t*. Daná struktura obsahuje všechna potřebná data pro IPFIX paket. Hlavičku IPFIX paketu uchovávající údaje o verzi formátu záznamu, identifikační číslo měřícího zařízení a podobně. Dále struktura obsahuje několik hlaviček pro datové bloky (Set Headers), struktury pro uložení šablon a paměť pro uložení dat jednotlivých datových toků. Exportování využívá dvě šablony. Jednu pro záznam datových toků využívajících protokol IP verze čtyři a druhou pro záznamy s protokolem IP verze šest. Důležitost rozdělení těchto protokolů spočívá v rozdílné velikosti IP adresy. IP adresa verze čtyři je třiceti dvou bitové číslo, ale u verze

šest se jedná o sto dvaceti osmi bitové číslo. Protože šablona definuje fixní velikost datového typu pro daný prvek, je nutné rozlišit záznamy rozdílných verzí IP protokolu. Šablony se přidávají do IPFIX paketu po uplynutí určitého časového limitu, jinak se exportuje paket obsahující pouze záznamy datových toků odlišených podle verze IP protokolu.

Identifikační číslo šablony = 256		Počet prvků = 14
0	Začátek datového toku = 154	Velikost prvku v bajtech = 8
0	Konec datového toku = 155	Velikost prvku v bajtech = 8
0	Počet paketů v datovém toku = 2	Velikost prvku v bajtech = 8
0	Velikost datového toku v bajtech = 1	Velikost prvku v bajtech = 8
0	Zdrojová IP adresa = 8	Velikost prvku v bajtech = 4
0	Cílová IP adresa = 12	Velikost prvku v bajtech = 4
0	Verze IP protokolu = 60	Velikost prvku v bajtech = 1
0	Protokol (UDP, TCP...) = 4	Velikost prvku v bajtech = 1
0	Zdrojový port = 7	Velikost prvku v bajtech = 2
0	Cílový port = 11	Velikost prvku v bajtech = 2
1	Průměrná mezipaketová mezera = 200	Velikost prvku v bajtech = 8
Číslo společnosti = 8000		
1	Maximální mezipaketová mezera = 201	Velikost prvku v bajtech = 8
Číslo společnosti = 8000		
1	Minimální mezipaketová mezera = 202	Velikost prvku v bajtech = 8
Číslo společnosti = 8000		
1	Rozptyl mezipaketových mezer = 203	Velikost prvku v bajtech = 4

Tabulka 3.1: Šablona pro záznamy s protokolem IP verze čtyři

Šablona pro záznamy s protokolem IP verze šest se mění pouze prvky s IP adresami a to následovně:

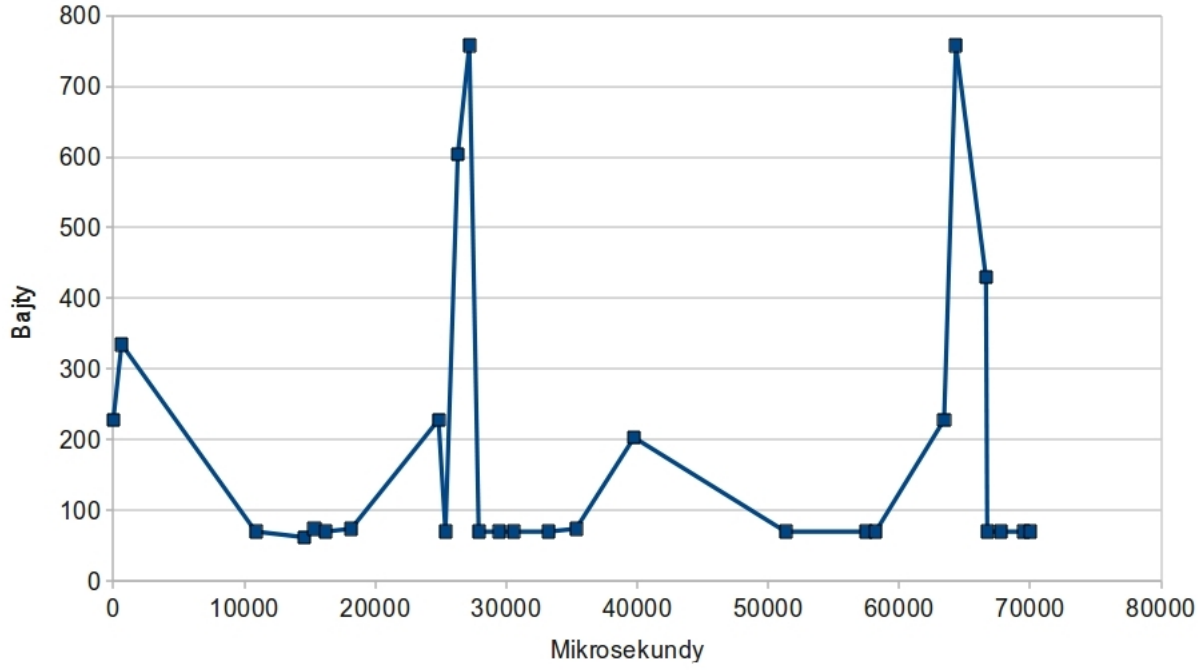
0	Zdrojová IP adresa = 27	Velikost prvku v bajtech = 16
0	Cílová IP adresa = 28	Velikost prvku v bajtech = 16

Tabulka 3.1: Prvky šablony pro uložení IP adresy verze 6

Samotná data jednotlivých datových toků se pomocí dvou struktur (*data_set_ip4_t*, *data_set_ip6_t*) ukládají do datového pole ve struktuře IPFIX paketu (*ipfix_pkt_t*). Jedná se o převod dat ze struktury *flow_t,* určené pro práci v rámci FTS, do formátu dat podle daných šablon pro export. IPFIX paket se odesílá, jakmile se naplní nebo při vypršení určitého časového intervalu.

3.1.9 Paketové statistiky

Paketové statistiky umožňují zachytávat velikost a čas příchodu každého paketu. Tato možnost je dostupná jen v textovém režimu. Jedná se o náročné měření jak pro procesor tak pro paměť, proto se výrazně nedoporučuje spouštět tento mód na rychlých sítích. Pro větší kontrolu nad zabranými zdroji při měření paketových statistik lze omezit maximální počet paketů pro daný datový tok, než se tok vypíše na výstup. Jakmile se tok vypíše na výstup, zachytí se opět dané maximum paketů pro daný tok. Výpis informací o jednotlivých paketech následuje za výpisem datového toku a to tak, že každý paket se vypíše na každý řádek zvlášť. Nejdříve se vypíše časová značka v mikrosekundách udávající příchod paketu, dále se vypíše velikost paketu v bajtech.



Obrázek 3.1: Ukázka hodnot z paketových statistik

3.1.10 Testování

Při testování Flow Time Statistic tool bylo potřeba ověřit správnost přiřazování časových značek, dále správnost přiřazených informací k daným datovým tokům a odesílaní dat pomocí IPFIX exportéru.

Správnost přiřazovaných informací a odesílaných dat se testovala pomocí paketového analyzátoru Wireshark. Během testování se odhalilo několik chyb. Při špatném formátování IPFIX paketu šlo v některých případech o špatné označení Whiresharku, jako například při odeslání IPFIX paketu naplněného pouze šablonami, kdy byl paket také označen za špatně formátovaný. Jinak se Whireshark ukázal jako velmi silný a užitečný nástroj pro testování IPFIX protokolu.

V rámci projektu bylo provedeno testování testovací skupinou. Ukázalo se, že při použití COMBO karty se na výstupu karty objevují pakety, které nebyly odeslány. Tento výsledek komplikuje další testování, jako například správnost neaktivního časového limitu a přesnost vypočtených časových statistik, protože do datových toků mohou být zařazeny vymyšlené pakety. Dále test, zda FTS stačí zpracovat všechny pakety při plném vytížení deseti gigabitové linky, je prozatím nesměrodatný, protože je výsledný počet paketů zvětšen o počet vymyšlených paketů a neví se, kolik paketů je neskutečných. Právě probíhají další testy v rámci projektu, které mají podat více informací o místě, kde se nachází chyba. Chyba může vznikat ve FTS nebo v samotné COMBO kartě, která je stále ve vývoji.

Při využití knihovny Pcap bylo zatížení procesoru a paměťové náročnosti prováděno na procesoru Intel Xeon 2.00GHz a k dispozici bylo 4GB operační paměti. Při rychlosti okolo 100Mb za sekundu bylo naměřeno zatížení okolo 15 až 16 procent procesoru a využitá paměť byla přibližně 89Mb. Při zvýšení přenosové rychlosti na 200Mbps stouplo zatížení na 29 procent, využití paměti zůstalo zhruba stejné. Okolo rychlosti 500Mbps bylo už zatížení procesoru takové, že se nepodařilo zachytit všechny pakety. Hranice ležela přibližně okolo 400Mbps, kdy byly ve většině případů zachyceny všechny pakety. Při srovnání výkonnosti textového výstupu a exportování pomocí IPFIX protokolu nebyl na daných rychlostech zjištěn zásadní rozdíl. Ovšem při zapnutí paketových statistik velmi záleželo na nastaveném limitu, přenosové rychlosti a rozmanitosti síťového provozu. Při testování FTS s Pcap knihovnou nebyly zaznamenány žádné vymyšlené pakety. Testování výkonnosti FTS při využití COMBO karty se prozatím odkládá z důvodu velkého množství vymyšlených paketů, které se objevují na výstupu FTS.

4 Závěr

Cílem bylo vytvořit nástroj pro měření časových statistik síťových spojení, schopný pracovat na rozhraních podporovaných knihovnou Pcap a na COMBO kartě s designem timestats+L7. Nástroj má dokázat exportovat záznamy pomocí IPFIX protokolu a měřit statistiky o čase příchodu a velikosti jednotlivých paketů.

Nástroj byl úspěšně implementován a dokáže pomocí knihovny Pcap zachytávat pakety a exportovat je pomocí IPFIX protokolu. Exportování IPFIX protokolu bylo ověřeno pomocí nástroje Whireshark, který je dostupný a má dobře implementovanou podporou IPFIX protokolu. Dále jsou dostupné módy pro textový a SQL výstup. Při měření pomocí COMBO karty bylo zjištěno, že se na výstupu objevují pakety, které nebyly zaslány na vstup. Prozatím není jasné, zda je chyba ve FTS nebo někde v COMBO kartě. Až se odhalí zdroj této chyby a chyba bude opravena, mohla by být zvetšena přesnost časových značek FTS při práci s COMBO kartou. FTS pracuje s časovými značkami s přesností na mikrosekundy, ale karta je schopná získat časové značky s přesností několika desítek nanosekund. Bylo by nutné přepracovat způsob práce FTS s časovými značkami a vyřešit v jakém formátu by se měl ukládat čas s přesností na nanosekundy.

Paketové statistiky jsou určeny na podrobnou analýzu paketů čistě na základě velikosti a času příchodu, s cílem identifikovat alespoň některé datové toky na základě paketových statistik. V ideálním případě by bylo možné rozeznat komunikaci na základě prvních několika paketů, přičemž by se dalo uvažovat i o nasazení na exportéru.

Literatura

[1] Simon Leinen: RFC3955 – Evaluation of Candidate Protocols for IP Flow Information Export [online]. Dostupný z WWW: <http://www.faqs.org/rfcs/rfc3955.html> (Květen 2010)

[2] Claise B.: RFC5101 - Specification of the Ip Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information [online].
Dostupný z WWW: <http://www.ietf.org/rfc/rfc5101.txt> (Květen 2010)

[3] Wikipedie: Otevřená encyklopedie: Network traffic measurement [online].
Dostupný z WWW: <http://en.wikipedia.org/wiki/Network_traffic_measurement>
(Květen 2010)

[4] C. Schmoll: RFC5471 – Guidelines for IP Flow Information Expor Testing [online].
Dostupný z WWW: <http://www.ietf.org/rfc/rfc5471.txt>
(Květen 2010)

[5] C. Chaudet, E. Fleury, I. Guérin Lassous, H. Rivano, M.-E. Voge: Optimal positioning of active and passive monitoring devices [online].
Dostupný z WWW:
<http://aeolus.ceid.upatras.gr/scientific-reports/copy_of_1st_year_reports/CFGR05b.pdf>
(Květen 2010)

[6] National Space-Based Positioning, Navigation, adn Timing Coordination Office: Global Positioning System [online].
Dostupný z WWW: <http://www.gps.gov/applications/timing/index.html> (Květen 2010)

[7] C. Sadasivan: RFC5470 – Architecture for IP Flow Information Export [online].
Dostupný z WWW: <http://www.ietf.org/rfc/rfc5470.txt>
(Květen 2010)

[8] Wikipedie: Otevřená encyklopedie: Netflow [online].
Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=Netflow&oldid=3773503>
(Květen 2010)

[9] Martin Malý: Přístup k internetu jako základní lidské právo? [online].
Dostupný z WWW:
<http://www.root.cz/clanky/pristup-k-internetu-jako-zakladni-lidske-pravo/>
(Květen 2010)

[10] Information Sciences Institute University of Southern California: Internet protocol [online].
Dostupný z WWW: <http://tools.ietf.org/html/rfc791>
(Květen 2010)

[11] Wikipedie: Otevřená encyklopedie: Packet analyzer [online].
Dostupný z WWW: <http://en.wikipedia.org/wiki/Packet_sniffer>
(Květen 2010)

[12] D. Harrington: An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks [online].
Dostupný z WWW: <http://tools.ietf.org/html/rfc3411>
(Květen 2010)

[16] Microsoft Corporation: Windows Management Instrumentation [online].
Dostupný z WWW: <http://msdn.microsoft.com/en-us/library/aa394582(v=VS.85).aspx>
(Květen 2010)

[17] J. Mogul: Pulse-Per-Second API for UNIX-like Operating Systems, Version 1.0 [online].
Dostupný z WWW: <http://tools.ietf.org/html/rfc2783>
(Květen 2010)

[18] Oldřich Mrázek: SNMP protokol a jeho využití [online].
Dostupný z WWW: <http://hw.cz/Produkty/ART957-SNMP-protokol-a-jeho-vyuziti.html>
(Květen 2010)

[19] Wikipedie: Otevřená encyklopedie: SQL [online].
Dostupný z WWW: <http://cs.wikipedia.org/wiki/SQL>
(Květen 2010)

[20] S. Deering, R. Hinden: Internet protocol, Version 6 (IPv6) Specification [online].
Dostupný z WWW: <http://tools.ietf.org/html/rfc2460>
(Květen 2010)

[21] David L. Mills: RFC1305 – Network Time Protocol (Version 3) Specification, Implementation and Analysis [online].
Dostupný z WWW: <http://www.faqs.org/rfcs/rfc1305.html> (Květen 2010)

[22] Rasmus Pagh, Flemming Friche Rodler: Cuckoo Hashing [online].
Dostupný z WWW: <http://cs.nyu.edu/courses/fall05/G22.3520-001/cuckoo-jour.pdf>
(Květen 2010)

[23] Tim Carstens: Programming with pcap [online].
Dostupný z WWW: <http://www.tcpdump.org/pcap.htm> (Květen 2010)

[24] Bret Mulvey: Hash Functions [online].
Dostupný z WWW: <http://home.comcast.net/~bretm/hash/> (Květen 2010)

[25] Falko Dressler, Georg Carle: High-Speed Network Monitoring and Analysis [online].
Dostupný z WWW:
<http://www.net.in.tum.de/fileadmin/bibtex/publications/papers/infocom2005_en.pdf>
(Květen 2010)

[26] Per Juvhaugen: Exportin IP flows using IPFIX [online].
Dostupný z WWW: <http://project.iu.hio.no/theses/pdf/master2007/per.pdf> (Květen 2010)

[27] NextCom:Vaša sieť pod kontrolou![online].
Dostupný z WWW:
<http://www.nextcom.sk/flowmon_monitorovanie_siete_uchovavanie_dat.xhtml>
(Květen 2010)

Seznam příloh

Příloha č. 1

Manuálová stránka FTS

NAME
fts - flow time statistic tool

SYNOPSIS
fts [-c limit] [-d hostname] [-h] [-H] [-i interface] [-I version]
[-m mode] [-n domain] [-p port] [-P protocol] [-q size]
[-t act:inact] [-v level] [-w file]

DESCRIPTION
The fts program serves as a software flow cache computing flow time statistics.

fts supports all network interfaces available through pcap library. It also supports reading data directly from fflowmon timestamps+L7 design using pseudo interface fflowmon.

OPTIONS
-c limit
Print out packet size and timestamp for each flow. Maximum packets can be limited by limit. Zero limit mean unlimited number of packets.

-d hostname
Set host name of collector for IPFIX exporter.

-h Print help message.

-H Print out column headers if available.

-i interface
Set interface to read data from. fts(1) is able to work with all pcap compatible interfaces. In this case fts(1) parses incomming packets and creates flow records in its flow cache.

On the other hand fts(1) also supports special pseudo interface fflowmon. In this case fts(1) reads preprocessed data from timestats+L7 fflowmon design and only calculates time statistics.

Interfaces can be set in the form if_nameN:M. N specifies card number and M specifies port to read data from. If port is skipped, data are read from all port of specified card. If neither card is set, default card number 0 is used.

-I version
Set version of IP protocol for export.
-m mode

Set output mode. Available modes are txt, sql or ipfix, txt output is default.

-n domain
Set domain name for ipfix exports.

-p port
Set port number of ipfix collector.

-P protocol
Set protocol name for ipfix exports. Available modes are tcp or udp.

-q size
Queue size in power of two. Default size is 19.

-t act:inact
Set active (act) and inactive (inact) timeout to specified values in seconds. Default value is 60:10, allowed range is 1-600 sec.

-v level
Set verbose output to level.

-w file
Write output to specified file instead of stdout.

OUTPUT
flow start, flow end, flow packets, flow bytes, l3 id, src->dst, l4 id, sport:dport, avg pkt diff, max pkt diff, min pkt diff, pkt dispersion.

flow start
Start of flow in usec.

flow end
End of flow in usec.

flow packets
Number of packets in flow.

flow bytes
Number of bytes in flow.

l3 id Number of ip version.br(taken from ip header: 4 = IPv4, 6 = IPv6...).

src->dst
Source and destination IP addresses.

l4 id Number of transport protocol.br(taken from ip header: 17 = UDP, 6 = TCP...).

sport:dport

Source and destination port numbers.

avg pkt diff
Average time gap between packets in flow..br(avg pkt diff = (flow end - flow start)/(flow packets - 1))

max pkt diff
Maximal time gap between packets selected from all diff's in flow..br(diff = packet time - previous packettime).

min pkt diff
Minimal time gap between packets selected from all diff's in flow..br(diff = packet time - previous packet time).

pkt dispersion
Dispersion of time gaps in flow.

Dispersion = sqrt((1/N) * (S-((P*P)/N))).

N - Number of gaps (flow packets - 1)..brP - Duration of flow (flow end - flow start)..brS - Sum of all squares of time gaps (time_diff2 += diff * diff).

EXAMPLES
fts -m sql -i fflowmon0:1
Read preprocessed data from port 1 at fflowmon design and print output in SQL format.

REPORTING BUGS
To report bug, go to [1]https://www.liberouter.org/bugtrack/

SEE ALSO
Manual pages:.TP fflowmon(1) main start up program for the Flexible Flowmon probe

Other resources:.TP [2]http://www.liberouter.org/fflowmon/ Flexible FlowMon probe web page

AUTHOR
Tomas Konir, Radek Krejci, Petr Kramolis.

REFERENCES
1. https://www.liberouter.org/bugtrack/
https://www.liberouter.org/bugtrack/

2. http://www.liberouter.org/fflowmon/
http://www.liberouter.org/fflowmon/